In-system Debugging of PCIe® Devices

Philippe Legros

PCI Express Architect

PLDA

Disclaimer

<u>Presentation Disclaimer</u>: All opinions, judgments, recommendations, etc. that are presented herein are the opinions of the presenter of the material and do not necessarily reflect the opinions of the PCI-SIG®.

Summary

- **Introduction**
- **Common problems in hardware**
- **Debug tools & techniques**
- **Conclusion**

Introduction

- **Debugging issues in a PCI Express® system is often challenging and very time consuming**
  - PCI Express is a complex protocol
  - At least two devices with each a controller and a PHY are involved
  - There is often little or no visibility
  - Problem can span over a long time period

Introduction

- **But fortunately PCI Express users benefit from a wide ecosystem of:**
  - Debug tools
  - Ways to increase visibility inside a device
  - Methods to troubleshoot issues
  - .. And also experts to assist with debugging

Summary

- Introduction
- **Common problems in hardware**
- Debug tools & techniques
- Conclusion

Common Problems in Hardware

- **Most of the problems seen in hardware are:**
  - Link is not training
  - Link is training but speed / width is not as expected
  - Missing/unexpected packets
  - Unexpected entry in Recovery

Link Training Issues

- **Common problems are:**
  - No link is ever established
    - Reset issue
    - Receiver detection issue
    - Link quality issue
  - Link width is not as expected
    - Lane(s) reliability issue

Link Speed Issue

- **There are two main scenarios:**
  - Device is not ready quickly enough after reset
    - Host system enumerates before device is ready and does not even attempt to train link at highest possible speed.
  - Speed change to 8.0 GT/s or 16 GT/s occurs but there is a timeout in equalization
    - Link automatically backs down to a lower speed (selected speed may depend on platform)

Traffic Issue

- **This is application specific so there are many possible symptoms:**
  - Missing TLPs
  - Duplicate TLPs
  - Unexpected traffic

Recovery

- **There are *many* reasons that can cause LTSSM to go to Recovery:**
  - Link quality issue
    - Encoding error
    - Framing error
  - Flow-control timeout
  - Device / link partner malfunction
  - Low-power entry/exit issue
  - …

Summary

- Introduction
- Common problems in hardware
- **Debug tools & techniques**
- Conclusion

Debug Tools and Techniques

- **Advanced error reporting**
- **Protocol analyzer**
- **Test interface**
- **Embedded analyzer**
- **Link reducers / Passive interposer card**

Advanced Error Reporting

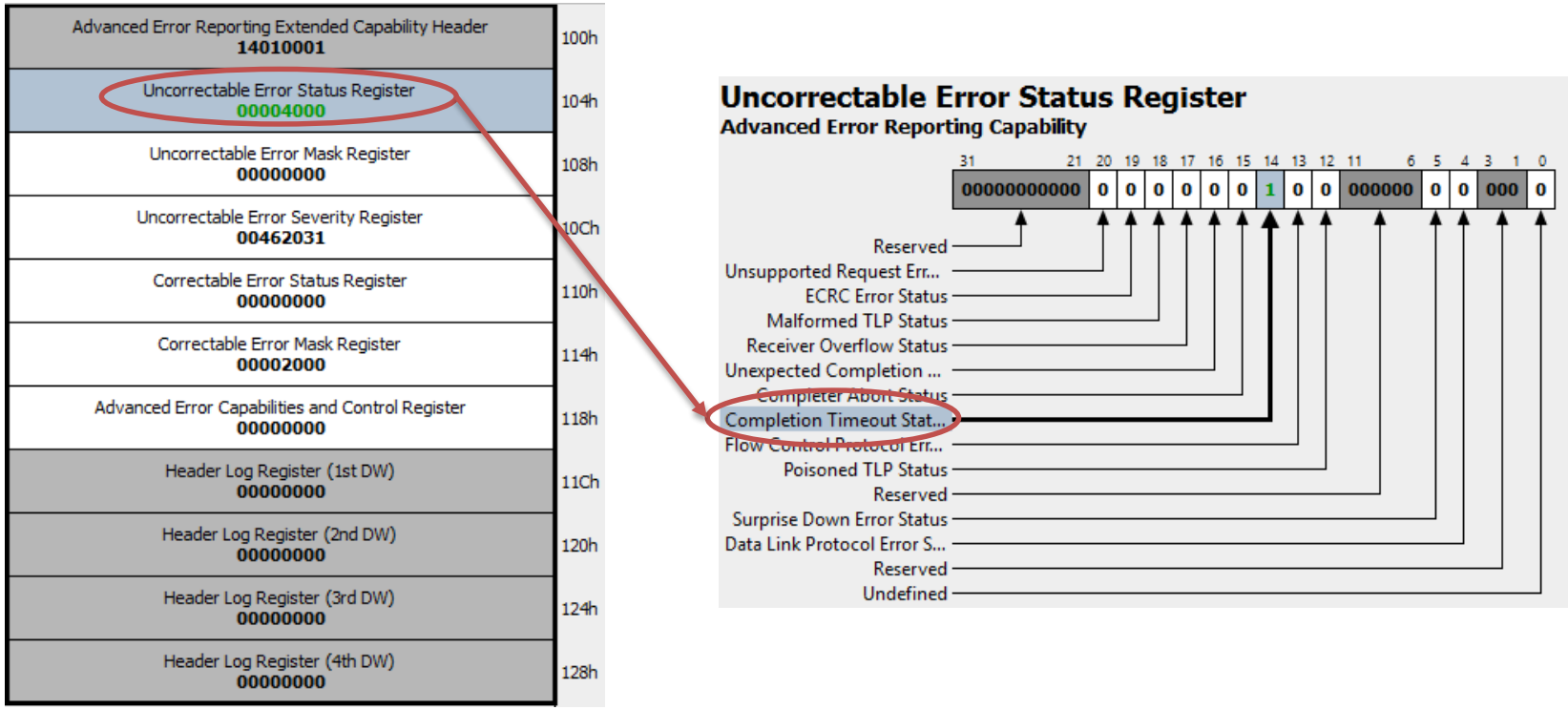
- **AER is an optional feature of PCIe® protocol that provides detailed information about errors that occur in a device:**
  - Separate status/mask/severity bit for each type of error
  - Offending TLP header is logged when applicable
  - Supports internal (i.e. application specific) errors

Advanced Error Reporting

- **Example:** “My device performs R/W DMA to system memory, but it stops transferring data after running for several minutes…”

Advanced Error Reporting

- **AER registers report that a Completion Timeout occurred:**



Advanced Error Reporting

- **So DMAs stopped because some completions are missing!**

- **Probable causes:**
  - Malformed read request
  - Design error in transmit or receive logic
  - …

Protocol Analyzer

- **This is the best known and most versatile tool to debug PCIe® issues**
  - Easy view of traffic
  - Powerful trigger conditions
  - Can filter out specific packets
  - Can record long sequences

Protocol Analyzer

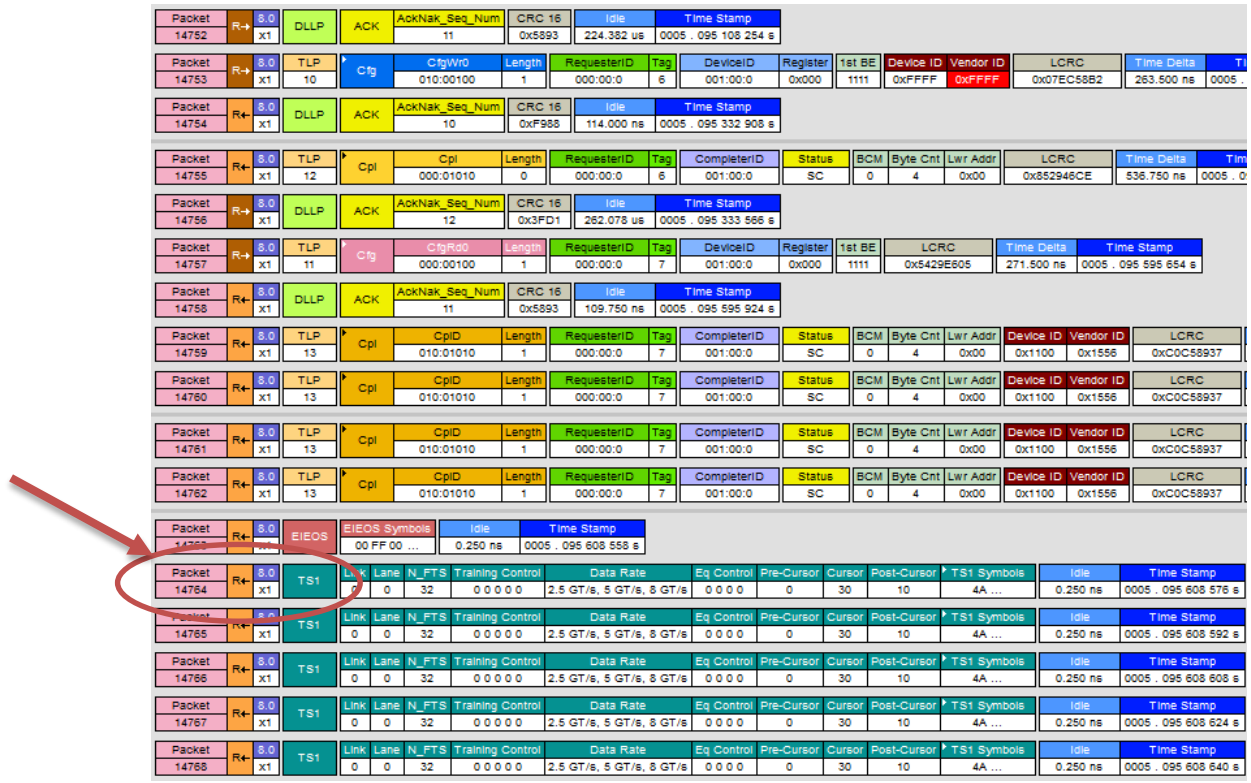
- **Example: Unexpected link retraining**

Packet				Link	Lane	N_FTS	Training Control	Data Rate	Eq Control	Pre-Cursor	Cursor	Post-Cursor	TS1 Symbols	Time Delta	Time Stamp
14793	R→	8.0 x1	TS1	0	0	255	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 1 0	0	20	4	4A ...	15.500 ns	0005 . 095 608 962 s
14794	R←	8.0 x1	TS1	0	0	32	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 0 0	0	30	10	4A ...	0.750 ns	0005 . 095 608 978 s
14795	R→	8.0 x1	TS1	0	0	255	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 1 0	0	20	4	4A ...	15.500 ns	0005 . 095 608 978 s
14796	R←	8.0 x1	TS1	0	0	32	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 0 0	0	30	10	4A ...	0.750 ns	0005 . 095 608 994 s
14797	R→	8.0 x1	TS1	0	0	255	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 1 0	0	20	4	4A ...	15.500 ns	0005 . 095 608 994 s
14798	R←	8.0 x1	TS1	0	0	32	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 0 0	0	30	10	4A ...	0.750 ns	0005 . 095 609 010 s
14799	R→	8.0 x1	TS2	0	0	255	0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 1 0	45 ...				15.500 ns	0005 . 095 609 010 s
14800	R←	8.0 x1	TS1	0	0	32	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 0 0	0	30	10	4A ...	0.750 ns	0005 . 095 609 026 s
14801	R→	8.0 x1	TS2	0	0	255	0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 1 0	45 ...				15.500 ns	0005 . 095 609 026 s
14802	R←	8.0 x1	TS1	0	0	32	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 0 0	0	30	10	4A ...	0.750 ns	0005 . 095 609 042 s
14803	R→	8.0 x1	TS2	0	0	255	0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 1 0	45 ...				15.500 ns	0005 . 095 609 044 s
14804	R←	8.0 x1	TS1	0	0	32	0 0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 0 0 0	0	30	10	4A ...	0.750 ns	0005 . 095 609 058 s
14805	R→	8.0 x1	TS2	0	0	255	0 0 0 0	2.5 GT/s, 5 GT/s, 8 GT/s	0 1 0	45 ...				15.500 ns	0005 . 095 609 060 s

So, how to proceed ?

Protocol Analyzer

- **First step is always to check which device initiates retraining**

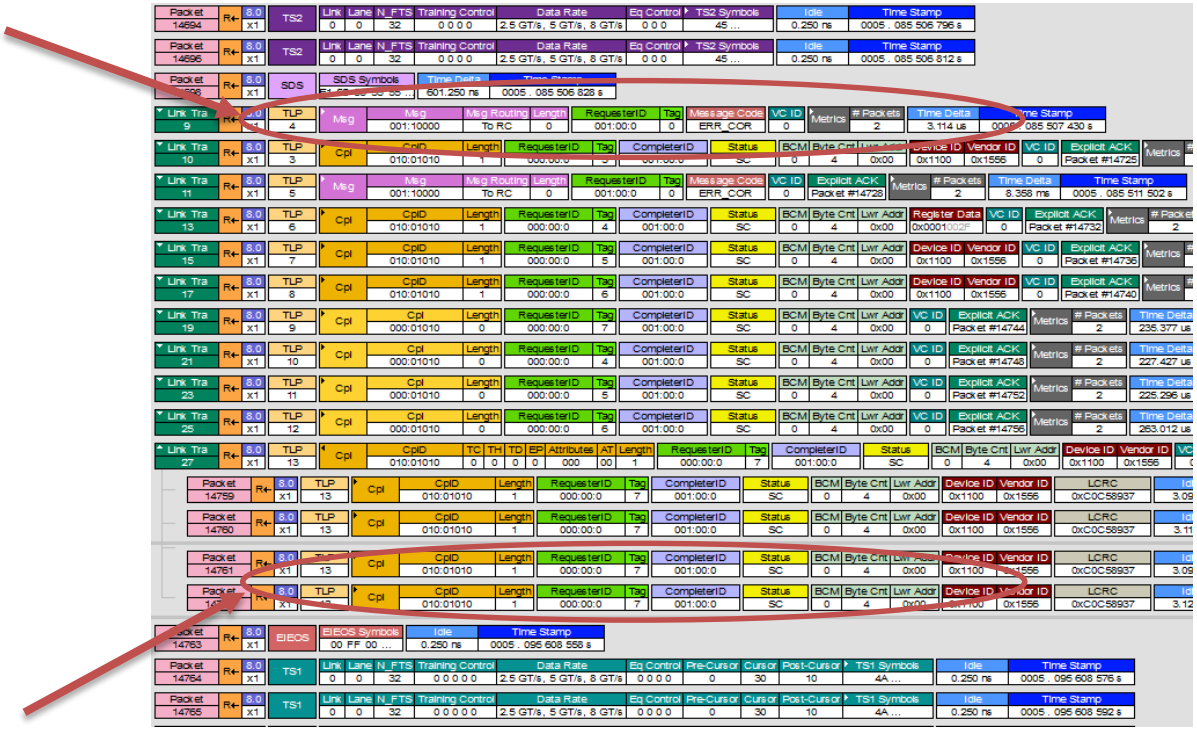


Protocol Analyzer

- **Then check if this device:**
  - Receives incorrect packets
  - Sends error messages (can be after retraining)
  - Is not receiving expected ACK, flow control DLLP, completion,…

Protocol Analyzer

- **Investigation shows that device transmitted error messages after previous link retrain**



Last TLP transmitted several times without ACK..

Protocol Analyzer

- **Protocol analyzers have limitations:**
  - They can show what the problem is, but not what the reason is !
  - They have limited ability to follow LTSSM and can show incorrect information.
  - They do not necessarily “see” all traffic :

    - Example : at 8.0 GT/s or higher some traffic can be missed (Equalization phase 0/1 for example) if analyzer does not lock on data quickly enough !

Test Interface

- **Having access to internal status signals can give precious hints:**
  - LTSSM
  - Error flags
  - Status information
  - Event counters
  - …

  These become even more precious if link cannot be established or is lost

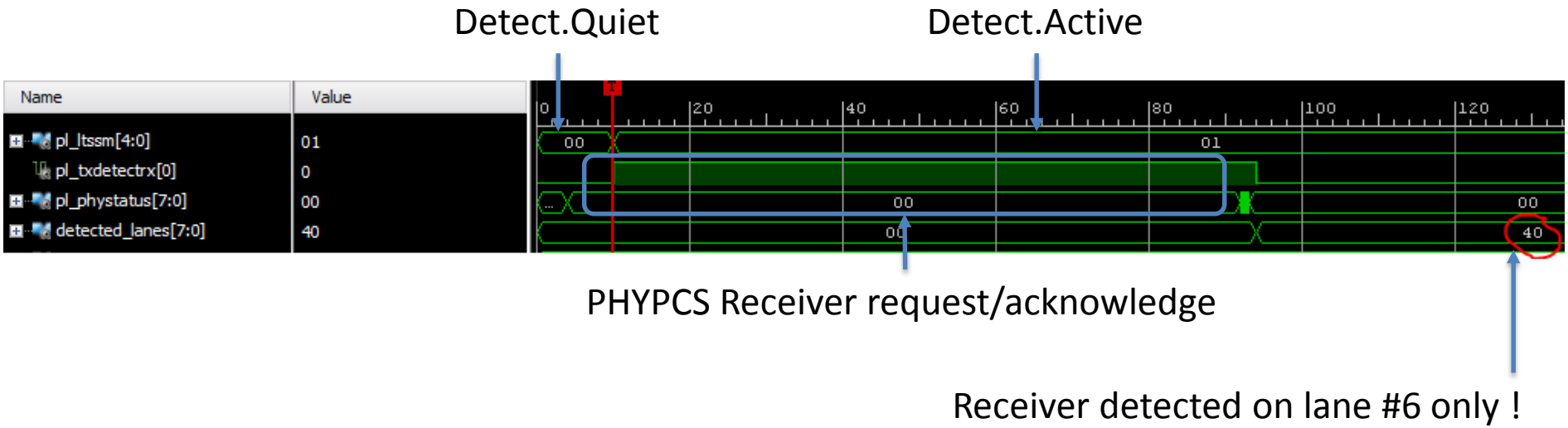
Embedded Analyzer

- **These tools can provide unparalleled insight on device:**
  - Internal states such as LTSSM
  - Error & event flags
  - Activity/traffic indicators
  - PHYPCS interface

- **Easy to use and widely available**

Embedded Analyzer

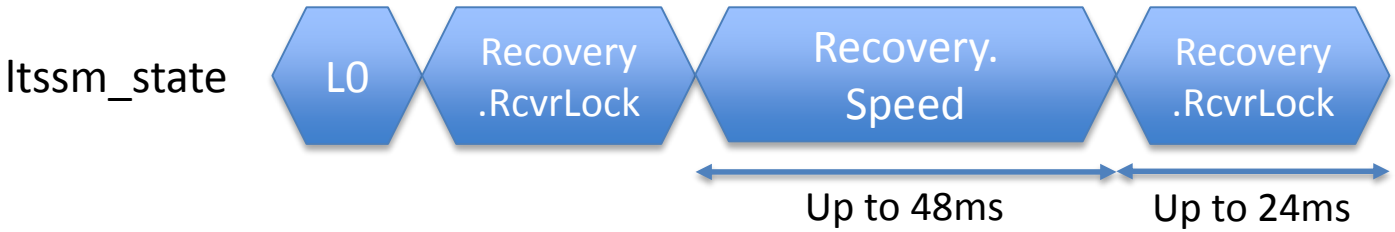
- **Example: Checking that receiver detection is behaving properly**



Embedded Analyzer

- **It is often difficult to have an overview of link training process:**
  - Transition between states can take several milliseconds
  - Timeout values in specification are 2…48 ms

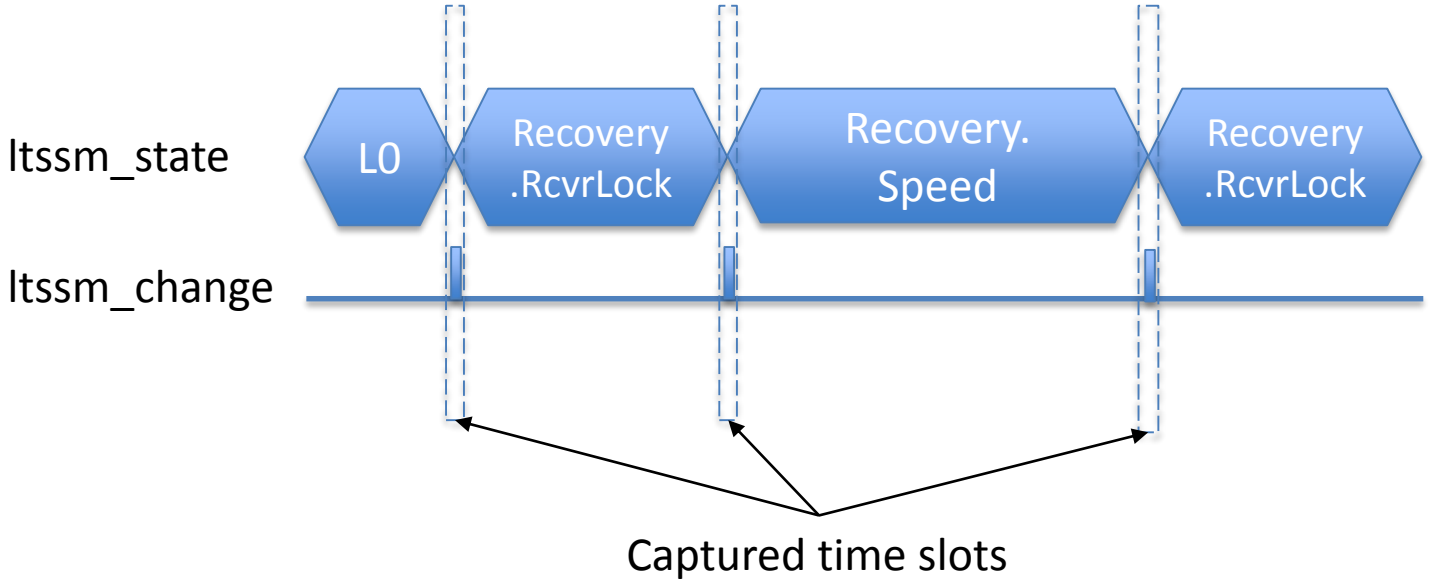
Example: Speed change



.. and embedded analyzer capture window is very limited

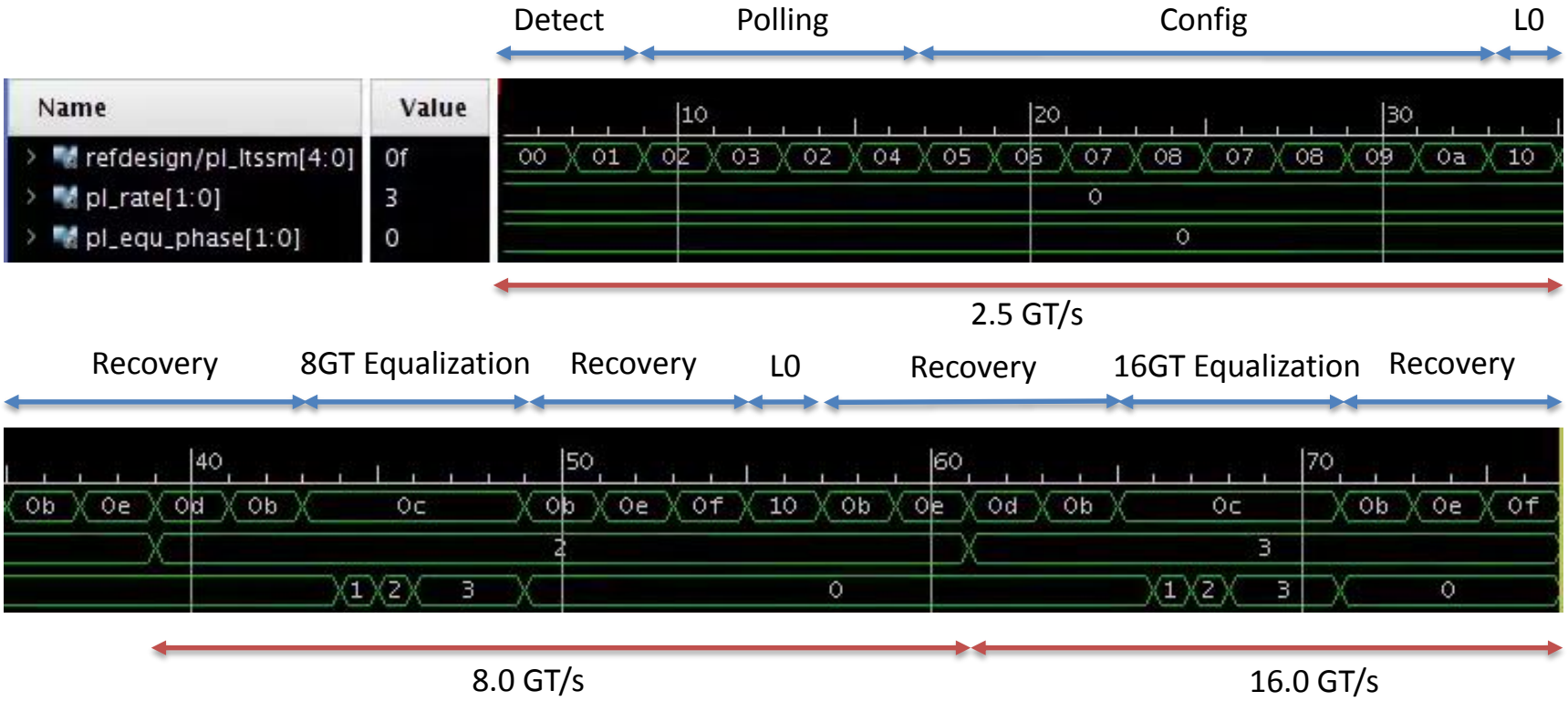
Embedded Analyzer

- **This can be solved by using a "storage qualifier" signal to tell embedded analyzer when to store data**



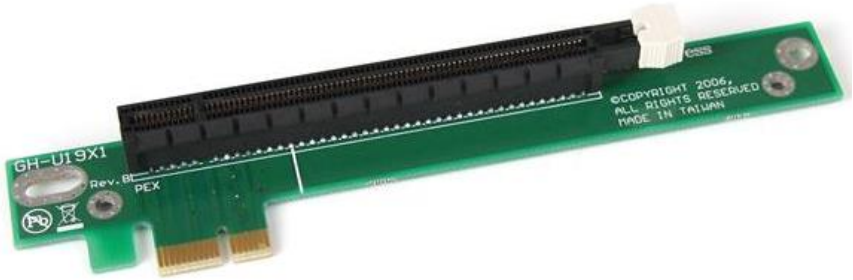
Embedded Analyzer

- **A complete link training sequence can then be captured with minimal sample buffer size:**



Link Reducer

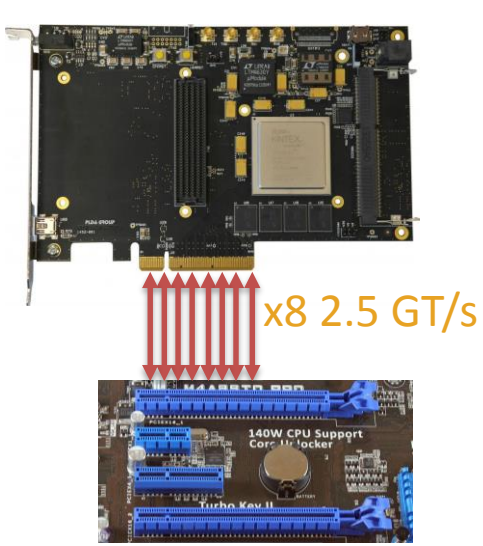
- **Sometimes the most inexpensive accessory can be very handy and help troubleshoot nasty issues..**



Should be used with caution: link reducers can sometimes deteriorate link quality and thus change device/system behavior

Link Reducer

- "I have a x8 8GT/s device, but it only comes up in x8 2.5GT/s in our platform.."
  - One quick experiment is to try with a link reducer:



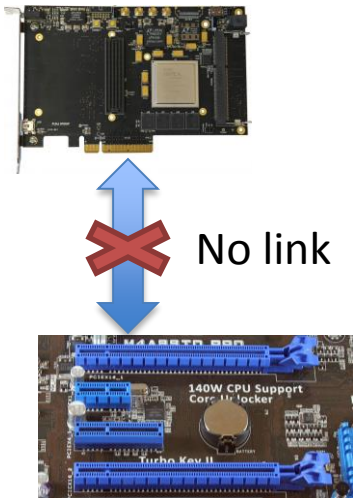
  - Link comes up at 8.0 GT/s in x1, showing that there is probably a link reliability issue on one or more lanes

Passive Interposer Card

- **Normally comes with a PCIe protocol analyzer, however it can also be a valuable stand-alone tool**
  - Can help troubleshoot some electrical issues

Passive Interposer Card

- "My device works in some systems, but in some others there is no link at all.."
  - A simple test is to plug it on a passive interposer card:



- Link comes up with an interposer, showing that there is an electrical problem (impedance,…)

Summary

- Introduction
- Common problems in hardware
- Debug tools & techniques
- **Conclusion**

Conclusion

- **There is not a single tool that can help troubleshoot all in-system issues**
  - It is important to understand which tool is the best suited for each type of problem
  - Having multiple tools / sources of information never hurts

Conclusion

- **Planning debugging during product development is absolutely critical!**
  - Enabling AER
  - Selecting key signals and adding them to a test port or embedded logic analyzer
  - Having ability to enable/disable features..

Each of these can be a life-saver

Thank you for attending the PCI-SIG Developers Conference 2017.

For more information please go to www.pcisig.com